UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/092,010	03/05/2002	Eric D. Bloch	LZLO-01001US0	7583

28554 7590 06/26/2008

VIERRA MAGEN MARCUS & DENIRO LLP
575 MARKET STREET SUITE 2500
SAN FRANCISCO, CA 94105

EXAMINER	
AILES, BENJAMIN A	

ART UNIT	PAPER NUMBER
2142	

MAIL DATE	DELIVERY MODE
06/26/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/092,010	BLOCH ET AL.	
	Examiner	Art Unit	
	BENJAMIN AILES	2142	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 February 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *See Continuation Sheet* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,4-11,13-17,19-24,27,28,30-33,36-58,60-62,64,65,67-70 and 73-82* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Disposition of Claims: Claims pending in the application are 1,4-11,13-17,19-24,27,28,30-33,36-58,60-62,64,65,67-70 and 73-82.

DETAILED ACTION

1. This action is in response to correspondence filed 01 February 2008.

2. Claims 1, 4-11 and 13-17, 19-24, 26-33, 35-58, 60-62, 64, 65, 67-70 and 73-80 remain pending.

Claim Objections

3. Claim 1 is objected to because of the following informalities:

-line 15, "…saud user device…" should be "…said user device…"

4. Claim 10 is objected to because of the following informalities:

-The claim has two periods.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claim 81 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

6. Regarding claim 81, the claim recites on line 2: "a name and/or format of the media content." The usage of "and/or" in the claim renders the claim indefinite because it is unclear if both the "name" and the "format" are required by the claim. For examination purposes, the limitation will be read in the alternative form. Further with respect to claim 81, the claim recites the "and/or" again. In accordance with above, the limitation will be read in the alternative form.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

9. Claims 1, 5-6, 8-10, 13, 21-24, 27, 28, 30-32, 37-40, 45-51, 53, 54, 67, 69, 70, 76 and 81 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tuli (US 7,068,381 B1) in view of Davis et al. (US 6,643,696 B2), hereinafter referred to as Davis.

10. Regarding claim 1, Tuli teaches a method for providing content, comprising the steps of:

receiving a request from a user device (col. 2, ll. 5-13, receive request at web server) for particular content, said request is received at a server (col. 2, ll. 5-13, receive request at web server);

accessing a mark-up language description of said particular content (col. 2, ll. 9-11, retrieve HTML), said mark-up language description includes one or more source files which

describe behavior of said particular content on a user interface of said user device (col. 2, ll. 11-12), said particular content includes data for rendering on said user interface (col. 2, ll. 1-14, graphics and text);

compiling said mark-up language description of said particular content (col. 2, ll. 11-12, browser translator), including said data, to create executable code for said user device, said step of compiling is performed at said server in response to said request (col. 2, ll. 5-13, fulfill request at web server); and

transmitting said executable code from said server to said user device (col. 2, ll. 40-42, data is sent back to user device), said user device provides said particular content via said user interface according to said one or more source files when said user device executes said executable code (col. 2, ll. 40-42, data is processed at user device).

Tuli teaches the accessing of a mark-up language description but not does not explicitly teach "accessing said data at said external data source based on said one or more source files which define said connection to said external data source, said server performs said accessing." However, in related art, Davis teaches on this aspect wherein a client device can send a request to a server for secondary content (col. 5, lines 54-58) and that the secondary content can be from an external data source (abstract, line 7). Tuli and Davis are analogous art because they are both from the same field of endeavor of computer systems. At the time of invention, it would have been obvious to one of ordinary skill in the art that Davis's method of calling an application from a previously downloaded webpage could be used with Tuli's method of compiling code at a server rather than at the client. After Davis's webpage is downloaded with Tuli's system, Davis's webpage would call the secondary application and Tuli's system would then proceed to

locate and compile that secondary application for presentation to the client. The motivation for doing so would have been to allow the users of Tuli's system to be able to utilize content of the type described in Davis on a thin-client device (col. 1, ll. 14-18). Therefore it would have been obvious to combine Davis with Tuli for the benefit of utilizing more complex content on a thin-client device.

11. Regarding claim 5, Tuli teaches the use of a browser translator to process web data (col. 1, ll. 32-36) but does not explicitly teach the compiling into ActionScript. Official notice is taken that it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to implement the browser translator to handle compilation into ActionScript because ActionScript was old and well known in the art. One of ordinary skill in the art would have been motivated to use ActionScript because of the common usage within web browsing.

12. Regarding claim 6, Tuli and Davis teach the method wherein said step of transmitting includes using HTTP to transmit said executable code via a network (fig. 1, Internet).

13. Regarding claim 8, Tuli and Davis teach the method comprising the steps of:

accessing media content, said particular content includes said media content (Tuli, col. 2, ll. 9-11, retrieve HTML);

providing a reference in said mark-up language description to a media file which contains said media content, said media file is external to said mark-up language description (Davis, col. 5, lines 54-58);

transmitting said media file with said executable code from said server to said user device (Tuli, col. 2, ll. 40-42, data is sent back to user device), a rendering entity at said user device

renders said media content on said user interface when said media file is referenced when said executable code is executed (Tuli, col. 2, ll. 40-42, data is processed at user device).

14. Regarding claim 9, Tuli teaches the use of a browser translator to process web data (col. 1, ll. 32-36) but does not explicitly teach the compiling into ActionScript. Official notice is taken that it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to implement the browser translator to handle compilation into ActionScript because ActionScript was old and well known in the art. One of ordinary skill in the art would have been motivated to use ActionScript because of the common usage within web browsing.

15. Regarding claim 10, Tuli and Davis teach the method further comprising: transforming said media file to an accepted format before said transmitting of said media file, said transforming is separate from said compiling (Tuli, col. 2, ll. 25-29).

16. Regarding claim 13, Tuli teaches that said particular content includes a first application (Tuli, col. 2, lines 5-13), and the steps of accessing a mark-up language description of content (Tuli, col. 2, lines 9-13), compiling said mark-up language description of content (Tuli, col. 2, lines 10-13), and transmitting said compiled mark-up language description of content to said client (Tuli, col. 2, ll. 40-42, data is sent back to user device). Tuli does not expressly teach the step of receiving a request from a client associated with said rendering entity for second content and that said second content includes a second application called by said first application. Davis teaches that a client device can send a request to a server for secondary content and that the second content can include a second application that is called by the first application (col. 5, lines 54-58). At the time of invention, it would have been obvious to one of ordinary skill in the art that Davis's method of calling an application from a previously downloaded webpage could be

used with Tuli's method of compiling code at a server rather than at the client. After Davis's webpage is downloaded with Tuli's system, Davis's webpage would call the secondary application and Tuli's system would then proceed to locate and compile that secondary application for presentation to the client. The motivation for doing so would have been to allow the users of Tuli's system to be able to utilize content of the type described in Davis on a thin-client device (col. 1, lines 59-61). Therefore it would have been obvious to combine Davis with Tuli for the benefit of utilizing more complex content on a thin-client device to obtain the invention as specified in claim 13.

17. Regarding claim 21, Tuli teaches a method for providing content, comprising the steps of:

receiving a request for content that includes data other than code (col. 2, ll. 5-13, receive request at web server), said data is for rendering on a user interface at a client, and said request is received at a server (col. 2, ll. 5-13, receive request at web server);

accessing a mark-up language description associated with said content at said server (col. 2, ll. 9-11, retrieve HTML);

acquiring said data from a data source external to and different than said server in response to said mark-up language description, said data is acquired by said server (col. 2, lines 46-47);

compiling said content at said server to create executable code, said content is based on said mark-up language description and said data, said executable code includes a representation of said data, said step of compiling is performed in response to said request (col. 2, ll. 11-12, browser translator); and

transmitting said executable code from said server to a client (col. 2, ll. 40-42, data is sent back to user device).

Tuli does not expressly disclose the step of receiving a request from said client for second content and that said second content includes a second application called by said first application. Davis teaches that a client device can send a request to a server for secondary content and that the second content can include a second application that is called by the first application (col. 5, lines 54-58). At the time of invention, it would have been obvious to one of ordinary skill in the art that Davis's method of calling an application from a previously downloaded webpage could be used with Tuli's method of compiling code at a server rather than at the client. After Davis's webpage is downloaded with Tuli's system, Davis's webpage would call the secondary application and Tuli's system would then proceed to locate and compile that secondary application for presentation to the client. The motivation for doing so would have been to allow the users of Tuli's system to be able to utilize content of the type described in Davis on a thin-client device (col. 1, lines 59-61). Therefore it would have been obvious to combine Davis with Tuli for the benefit of utilizing more complex content on a thin-client device to obtain the invention as specified in claim 21.

18. Regarding claim 22, Tuli and Davis teach that said request is from said client (Tuli, col. 2, ll. 9-10).

19. Regarding claim 23, Tuli and Davis teach that said executable code implements a user interface that provides access to said data (Tuli, col. 2, ll. 5-13, mini-browser).

20. Regarding claim 24, Tuli teaches the use of a browser translator to process web data (col. 1, ll. 32-36) but does not explicitly teach the compiling into ActionScript. Official notice is taken

that it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to implement the browser translator to handle compilation into ActionScript because ActionScript was old and well known in the art. One of ordinary skill in the art would have been motivated to use ActionScript because of the common usage within web browsing.

21. Regarding claim 27, Tuli and Davis teach the method further comprising the steps of:

accessing media content (col. 2, ll. 9-11, retrieve HTML);

transforming said media content to an accepted format, said transforming is separate from said compiling (Tuli, col. 2, ll. 25-29); and

transmitting said transformed media content with said executable code from said server to said client (Tuli, col. 2, ll. 40-42, data is processed at user device).

22. Regarding claim 37, Tuli teaches a method for providing content, comprising the steps of:

receiving a request from a user device (col. 2, ll. 5-13, receive request at web server) for particular content, said request is received at a server (col. 2, ll. 5-13, receive request at web server);

accessing a mark-up language description of said particular content (col. 2, ll. 9-11, retrieve HTML), said mark-up language description includes one or more source files which describe behavior of said particular content on a user interface of said user device (col. 2, ll. 11-12), said particular content includes data for rendering on said user interface (col. 2, ll. 1-14, graphics and text);

compiling said mark-up language description of said particular content (col. 2, ll. 11-12, browser translator), including said data, to create executable code for said user device, said step

of compiling is performed at said server in response to said request (col. 2, ll. 5-13, fulfill request at web server); and

transmitting said executable code from said server to said user device (col. 2, ll. 40-42, data is sent back to user device), said user device provides said particular content via said user interface according to said one or more source files when said user device executes said executable code (col. 2, ll. 40-42, data is processed at user device).

Tuli teaches the accessing of a mark-up language description but not does not explicitly teach "accessing said data at said external data source based on said one or more source files which define said connection to said external data source, said server performs said accessing." However, in related art, Davis teaches on this aspect wherein a client device can send a request to a server for secondary content (col. 5, lines 54-58) and that the secondary content can be from an external data source (abstract, line 7). Tuli and Davis are analogous art because they are both from the same field of endeavor of computer systems. At the time of invention, it would have been obvious to one of ordinary skill in the art that Davis's method of calling an application from a previously downloaded webpage could be used with Tuli's method of compiling code at a server rather than at the client. After Davis's webpage is downloaded with Tuli's system, Davis's webpage would call the secondary application and Tuli's system would then proceed to locate and compile that secondary application for presentation to the client. The motivation for doing so would have been to allow the users of Tuli's system to be able to utilize content of the type described in Davis on a thin-client device (col. 1, ll. 14-18). Therefore it would have been obvious to combine Davis with Tuli for the benefit of utilizing more complex content on a thin-client device.

23. Regarding claim 38, Tuli and Davis teach wherein said request is from said client (Tuli, col. 2, ll. 9-10).

24. Regarding claim 39, Tuli and Davis teach wherein said executable code implements said user interface, said user interface provides access to said data (Tuli, col. 2, ll. 50-55).

25. Regarding claim 40, Tuli and Davis teach the method further comprising the steps of:

accessing media content (Tuli, col. 2, ll. 9-11, retrieve HTML);

providing a reference to said media content in said executable code (Tuli, col. 2, lines 47-50); and

transmitting said media content with said executable code from said server to said rendering entity (col. 2, ll. 40-42, data is sent back to user device).

26. Regarding claim 45, Tuli teaches an apparatus, comprising:

one or more storage devices (col. 2, ll. 6, host computer); and

one or more processors in communication with said one or more storage devices (col. 2, ll. 6, host computer), said one or more processors perform a method comprising the steps of:

receiving a request for particular content, said request is received at a server (col. 2, ll. 5-13, receive request at web server), said request is from a client, said client includes a browser and a rendering engine that is different than said browser but operates in connection with said browser (col. 2, ll. 21-25);

accessing a mark-up language description of said particular content (col. 2, ll. 9-11, retrieve HTML), said mark-up language description includes one or more source files which describe behavior of said particular content on a user interface of said user device (col. 2, ll. 11-

12), said particular content includes data for rendering on said user interface (col. 2, ll. 1-14, graphics and text);

compiling said mark-up language description of said particular content (col. 2, ll. 11-12, browser translator), including said data, to create executable code for said user device, said step of compiling is performed at said server in response to said request (col. 2, ll. 5-13, fulfill request at web server); and

transmitting said executable code from said server to said user device (col. 2, ll. 40-42, data is sent back to user device), said user device provides said particular content via said user interface according to said one or more source files when said user device executes said executable code (col. 2, ll. 40-42, data is processed at user device).

Tuli teaches the accessing of a mark-up language description but not does not explicitly teach the definition of a connection to an external data source for data wherein the external data source is external to the server. However, in related art, Davis teaches on this aspect wherein a client device can send a request to a server for secondary content (col. 5, lines 54-58) and that the secondary content can be from an external data source (abstract, line 7). Tuli and Davis are analogous art because they are both from the same field of endeavor of computer systems. At the time of invention, it would have been obvious to one of ordinary skill in the art that Davis's method of calling an application from a previously downloaded webpage could be used with Tuli's method of compiling code at a server rather than at the client. After Davis's webpage is downloaded with Tuli's system, Davis's webpage would call the secondary application and Tuli's system would then proceed to locate and compile that secondary application for presentation to the client. The motivation for doing so would have been to allow the users of

Tuli's system to be able to utilize content of the type described in Davis on a thin-client device (col. 1, lines 59-61). Therefore it would have been obvious to combine Davis with Tuli for the benefit of utilizing more complex content on a thin-client device.

27. Regarding claim 46, Tuli and Davis teach an apparatus wherein:

said particular content includes data stored at said source, said accessing first code includes accessing said data at said source ((col. 2, ll. 9-11, retrieve HTML)); and

said data is compiled to executable code during said step of compiling (Tuli, col. 2, ll. 9-13, browser translator).

28. Regarding claim 47, Tuli and Davis teach the steps of:

accessing media content, said particular content includes said media content, at least one of said elements identifies said media content (col. 2, ll. 9-11, retrieve HTML);

transforming said media content to an accepted format, said transforming is separate from said compiling (Tuli, col. 2, ll. 25-29); and

transmitting said transformed media content with said executable code to said client for rendering of said transformed media content by said rendering engine (Tuli, col. 2, ll. 40-42, data is processed at user device).

29. Regarding claim 48, Tuli discloses an apparatus, comprising:

one or more storage devices (col. 2, ll. 6, host computer); and

one or more processors in communication with said one or more storage devices (col. 2, ll. 6, host computer), said one or more processors perform a method comprising the steps of:

receiving a request for particular content, said request is received at a server (col. 2, ll. 5-13, receive request at web server), said request is from a client, said client

includes a browser and a rendering engine that is different than said browser but operates in connection with said browser (col. 2, ll. 21-25);

accessing a mark-up language description of said particular content (col. 2, ll. 9-11, retrieve HTML), said mark-up language description includes one or more source files which describe behavior of said particular content on a user interface of said user device (col. 2, ll. 11-12), said particular content includes data for rendering on said user interface (col. 2, ll. 1-14, graphics and text);

compiling said mark-up language description of said particular content (col. 2, ll. 11-12, browser translator), including said data, to create executable code for said user device, said step of compiling is performed at said server in response to said request (col. 2, ll. 5-13, fulfill request at web server); and

transmitting said executable code from said server to said user device (col. 2, ll. 40-42, data is sent back to user device), said user device provides said particular content via said user interface according to said one or more source files when said user device executes said executable code (col. 2, ll. 40-42, data is processed at user device).

Tuli discloses that said particular content includes a first application (col. 2, lines 45-46), and the steps of accessing a mark-up language description of content (col. 2, lines 46-47, 57), compiling said mark-up language description of content (col. 2, lines 47-50), and transmitting said compiled mark-up language description of content to said client (col. 2, lines 50-51). Tuli does not expressly disclose the step of receiving a request from said client for second content and that said second content includes a second application called by said first application. Davis teaches that a client device can send a request to a server for secondary content and that the

second content can include a second application that is called by the first application (col. 5, lines 54-58). At the time of invention, it would have been obvious to one of ordinary skill in the art that Davis's method of calling an application from a previously downloaded webpage could be used with Tuli's method of compiling code at a server rather than at the client. After Davis's webpage is downloaded with Tuli's system, Davis's webpage would call the secondary application and Tuli's system would then proceed to locate and compile that secondary application for presentation to the client. The motivation for doing so would have been to allow the users of Tuli's system to be able to utilize content of the type described in Davis on a thin-client device (col. 1, lines 59-61). Therefore it would have been obvious to combine Davis with Tuli for the benefit of utilizing more complex content on a thin-client device to obtain the invention as specified in claim 48.

30. Regarding claim 49, Tuli discloses that said executable code implements a user interface that provides access to said data (col. 2, lines 49-55).

31. Regarding claim 50, Tuli discloses the steps of:

said data includes media content (col. 4, ll. 18-22).

32. Regarding claim 51, Tuli and Davis teach the method wherein said data is media data (Tuli, col. 4, ll. 18-21).

33. Regarding claim 53, Tuli and Davis teach the method wherein said executable code comprises one or more binary files (Tuli, col. 4, ll. 18-21).

34. Regarding claim 54, Tuli and Davis teach the method wherein said executable code comprises at least one of object code and byte code (Tuli, col. 4, ll. 18-21).

35. Regarding claim 67, Tuli and Davis teach the method wherein:

said markup language description comprises elements which are identified by markup language tags (Tuli, col. 2, ll. 10-11, HTML); and

said elements comprise at least one element which references said connection to said external data source (Davis, (col. 5, lines 54-58)).

36. Regarding claim 69, Tuli and Davis teach a method wherein said compiling comprises parsing said markup language description to obtain first and second types of elements, providing said first and second types of elements to first and second compiling modules, respectively, to obtain first and second object code, respectively, and assembling said first and second object code into a single executable (Tuli, col. 2, ll. 9-13, browser translator).

37. Regarding claim 70, Tuli the method wherein said first type of element defines at least one of a visual appearance of said content (col. 4, ll. 16-22). Tuli teaches the accessing of a mark-up language description but not does not explicitly teach the definition of a connection to an external data source for data wherein the external data source is external to the server. However, in related art, Davis teaches on this aspect wherein a client device can send a request to a server for secondary content (col. 5, lines 54-58) and that the secondary content can be from an external data source (abstract, line 7). Tuli and Davis are analogous art because they are both from the same field of endeavor of computer systems. At the time of invention, it would have been obvious to one of ordinary skill in the art that Davis's method of calling an application from a previously downloaded webpage could be used with Tuli's method of compiling code at a server rather than at the client. After Davis's webpage is downloaded with Tuli's system, Davis's webpage would call the secondary application and Tuli's system would then proceed to locate and compile that secondary application for presentation to the client. The motivation for

doing so would have been to allow the users of Tuli's system to be able to utilize content of the type described in Davis on a thin-client device (col. 1, lines 59-61). Therefore it would have been obvious to combine Davis with Tuli for the benefit of utilizing more complex content on a thin-client device.

38. Regarding claim 76, Tuli and Davis teach a method wherein said accepted format comprises at least one of a JPEG format and a GIF format (Tuli, col. 4, ll. 18-21).

39. Regarding claim 81, Tuli and Davis teach a method further comprising: providing an object in the executable code which identifies a name or format of the media content, the name or format is provided via the user interface when said media content is rendered (Tuli, col. 4, ll. 18-22).

40. Claims 4, 7, 28, 30-33, 36, 41-44, 52, 55-58, 60-62, 64, 65, 73, 77, 78, 80 and 82 rejected under 35 U.S.C. 103(a) as being unpatentable over Tuli and Davis in view of Rubin et al. (US 6,701,522 B1), hereinafter referred to as Rubin.

41. Regarding claim 4, Tuli teaches a user device including a rendering entity (col. 1, line 66 – col. 2, ll. 4, operating system with a mini-browser) but does not explicitly teach "said rendering entity is a plug-in to said browser, said plug-in is embedded in said browser before said request, and said rendering entity executes said executable code." However, in related art, Rubin teaches on this limitation wherein a portable device is equipped with plug-in installed within a user device's web browser (col. 7, ll. 18-21). One of ordinary skill in the art at the time of the applicant's invention would have found it obvious to utilize the plug-in as taught by Rubin in combination with the portable device web browser taught by Tuli. One of ordinary skill would

have been motivated to combine Rubin with Tuli because plug-ins are auxiliary programs added to web browsers that provide them with new functionality (Rubin, col. 7, ll. 21-23).

42. Regarding claim 7, Tuli teaches a user device including a rendering entity (col. 1, line 66 – col. 2, ll. 4, operating system with a mini-browser) but does not explicitly teach "said rendering entity is a plug-in to said browser, said plug-in is embedded in said browser before said request, and said rendering entity executes said executable code." However, in related art, Rubin teaches on this limitation wherein a portable device is equipped with plug-in installed within a user device's web browser (col. 7, ll. 18-21). One of ordinary skill in the art at the time of the applicant's invention would have found it obvious to utilize the plug-in as taught by Rubin in combination with the portable device web browser taught by Tuli. One of ordinary skill would have been motivated to combine Rubin with Tuli because plug-ins are auxiliary programs added to web browsers that provide them with new functionality (Rubin, col. 7, ll. 21-23).

43. Regarding claim 28, Tuli teaches one or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising the steps of:

receiving a request for particular content from a browser (col. 2, ll. 5-13, receive request at web server), said request is received at a server (col. 2, ll. 5-13, receive request at web server);

accessing a mark-up language description of said particular content (col. 2, lines 46-47, 57), said mark-up language description references a media file comprising at least one of audio, video and a movie (col. 4, ll. 16-22);

compiling said mark-up language description of said particular content to create executable code (col. 2, ll. 11-12, browser translator), said executable code provides said particular content, said step of compiling is performed at said server in response to said request (col. 2, ll. 5-13, fulfill request at web server); and

transmitting said executable code and said media file from said server, said media file is not compiled (col. 2, ll. 40-42, data is processed at user device).

Tuli teaches a user device including a rendering entity (col. 1, line 66 – col. 2, ll. 4, operating system with a mini-browser) but does not explicitly teach "said rendering entity is a plug-in to said browser, said plug-in is embedded in said browser before said request, and said rendering entity executes said executable code." However, in related art, Rubin teaches on this limitation wherein a portable device is equipped with plug-in installed within a user device's web browser (col. 7, ll. 18-21). One of ordinary skill in the art at the time of the applicant's invention would have found it obvious to utilize the plug-in as taught by Rubin in combination with the portable device web browser taught by Tuli. One of ordinary skill would have been motivated to combine Rubin with Tuli because plug-ins are auxiliary programs added to web browsers that provide them with new functionality (Rubin, col. 7, ll. 21-23).

44. Regarding claim 30, Tuli and Rubin teach that said executable code implements a user interface that provides access to said particular content (Tuli, col. 2, ll. 5-13, mini-browser).

45. Regarding claim 31, Tuli and Rubin teach:

said particular content includes data (Tuli, col. 2, ll. 9-11, retrieve HTML); and

said data is compiled to executable code during said step of compiling (Tuli, col. 2, ll. 9-11, browser translator).

46. Regarding claim 32, Tuli and Rubin teach that said method further comprises the steps of: transforming said media file to an accepted format before transmitting said media file, said transforming is separate from said compiling (Tuli, col. 2, ll. 25-29).

47. Regarding claim 33, Tuli teaches one or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising the steps of:

receiving a request for particular content from a browser (col. 2, ll. 5-13, receive request at web server), said request is received at a server (col. 2, ll. 5-13, receive request at web server);

accessing a mark-up language description of said particular content (col. 2, lines 46-47, 57), said mark-up language description references a media file comprising at least one of audio, video and a movie (col. 4, ll. 16-22);

compiling said mark-up language description of said particular content to create executable code (col. 2, ll. 11-12, browser translator), said executable code provides said particular content, said step of compiling is performed at said server in response to said request (col. 2, ll. 5-13, fulfill request at web server); and

transmitting said executable code and said media file from said server, said media file is not compiled (col. 2, ll. 40-42, data is processed at user device).

Tuli teaches a user device including a rendering entity (col. 1, line 66 – col. 2, ll. 4, operating system with a mini-browser) but does not explicitly teach "said rendering entity is a plug-in to said browser, said plug-in is embedded in said browser before said request, and said rendering entity executes said executable code." However, in related art, Rubin teaches on this

limitation wherein a portable device is equipped with plug-in installed within a user device's web browser (col. 7, ll. 18-21). One of ordinary skill in the art at the time of the applicant's invention would have found it obvious to utilize the plug-in as taught by Rubin in combination with the portable device web browser taught by Tuli. One of ordinary skill would have been motivated to combine Rubin with Tuli because plug-ins are auxiliary programs added to web browsers that provide them with new functionality (Rubin, col. 7, ll. 21-23).

48. Regarding claim 36, Tuli and Rubin teach the steps of:

accessing media content, said particular content includes said media content (Tuli, col. 2, ll. 9-11, retrieve HTML);

transforming said media content to an accepted format, said transforming is separate from said compiling (Tuli, col. 2, ll. 25-29); and

adding said transformed media content to said executable code (Tuli, col. 2, ll. 25-29).

49. Regarding claim 41, Tuli teaches one or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising the steps of:

receiving a request for particular content from a browser (col. 2, ll. 5-13, receive request at web server), said request is received at a server (col. 2, ll. 5-13, receive request at web server);

accessing a mark-up language description of said particular content (col. 2, lines 46-47, 57), said mark-up language description references a media file comprising at least one of audio, video and a movie (col. 4, ll. 16-22);

compiling said mark-up language description of said particular content to create executable code (col. 2, ll. 11-12, browser translator), said executable code provides said particular content, said step of compiling is performed at said server in response to said request (col. 2, ll. 5-13, fulfill request at web server); and

transmitting said executable code and said media file from said server, said media file is not compiled (col. 2, ll. 40-42, data is processed at user device).

Tuli teaches a user device including a rendering entity (col. 1, line 66 – col. 2, ll. 4, operating system with a mini-browser) but does not explicitly teach "said rendering entity is a plug-in to said browser, said plug-in is embedded in said browser before said request, and said rendering entity executes said executable code." However, in related art, Rubin teaches on this limitation wherein a portable device is equipped with plug-in installed within a user device's web browser (col. 7, ll. 18-21). One of ordinary skill in the art at the time of the applicant's invention would have found it obvious to utilize the plug-in as taught by Rubin in combination with the portable device web browser taught by Tuli. One of ordinary skill would have been motivated to combine Rubin with Tuli because plug-ins are auxiliary programs added to web browsers that provide them with new functionality (Rubin, col. 7, ll. 21-23).

50. Regarding claim 42, Tuli and Rubin teach said executable code implements a user interface that provides access to said particular content (Tuli, col. 2, ll. 49-55).

51. Regarding claim 43, Tuli and Rubin teach wherein:

said particular content includes data (Tuli, col. 4, ll. 18-21); and

said data is compiled to executable code during said step of compiling (Tuli, col. 4, ll. 18-21).

52. Regarding claim 44, Tuli and Rubin teach wherein said particular content includes at least one of audio, video and a movie (Tuli, col. 4, ll. 16-22).

53. Regarding claim 52, Tuli and Rubin teach a method wherein:

said request is received at said server from said user device and includes an indication that identifies a type of said rendering entity from a group of rendering entities (col. 2, ll. 5-13, receive request at web server); and

said compiling includes creating said executable code specific for said type of rendering entity in response to said indication (Tuli, col. 2, ll. 5-13, mini-browser).

54. Regarding claim 55, Tuli and Rubin teach the one or more processor readable storage devices wherein said first comprises elements which are identified by markup language tags (Tuli, col. 2, ll. 9-10, use of HTML).

55. Regarding claim 56, Tuli and Rubin teach one or more processor readable storage devices wherein: at least one of said elements define a view template of a user interface element, said view template is instantiated when said executable code is executed by said rendering entity (Tuli, col. 2, ll. 19-24).

56. Regarding claim 57, Tuli and Rubin teach one or more processor readable storage devices wherein said elements comprise at least one element which defines a view class which supplies default properties, behavior, and child views which the view template instantiates, the child views are associated with a parent view (Tuli, col. 2, ll. 19-24).

57. Regarding claim 58, Tuli and Rubin teach one or more processor readable storage devices wherein at least one of said elements references a media file comprising at least one of audio, video and a movie (Tuli, col. 4, ll. 16-22).

58. Regarding claim 60, Tuli and Rubin teach at least one of said elements references a media file that contains an animation (Tuli, col. 4, ll. 16-22).

59. Regarding claim 61, Tuli and Rubin teach at least one of said elements references a media file that contains a movie (Tuli, col. 4, ll. 16-22).

60. Regarding claim 62, Tuli teaches the use of a browser to display data on a user's device (col. 1, ll. 38-41) but does not explicitly teach the displaying of .SWF files. Official notice is taken that it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to implement the browser to display .SWF files because .SWF files were old and well known in the art. One of ordinary skill in the art would have been motivated to use .SWF files because of the common usage within web browsing.

61. Regarding claim 64, Tuli and Rubin teach wherein at least one of said elements provides an inline definition of formatted text (Tuli, col. 2, ll. 59-63).

62. Regarding claim 65, Tuli and Rubin teach wherein at least one of said elements provides an inline definition of vector graphics (Tuli, col. 2, ll. 59-63).

63. Regarding claim 73, Tuli teaches the use of a browser to display data on a user's device (col. 1, ll. 38-41) but does not explicitly teach the usage of a Flash player. Official notice is taken that it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to implement the browser to utilize a Flash player because Flash players were old and well known in the art. One of ordinary skill in the art would have been motivated to use a Flash player because of the common usage within web browsing.

64. Regarding claim 77, Tuli and Rubin teach wherein said elements comprises elements which define script code, said script code specifies a visual appearance of said user interface (Tuli, col. 2, ll. 50-55).

65. Regarding claim 78, Tuli and Rubin teach wherein said elements comprises elements which define script code, said script code specifies an application logic of said mark-up language description (Tuli, col. 2, ll. 50-55).

66. Regarding claim 80, Tuli teaches the use of a browser to display data on a user's device (col. 1, ll. 38-41) but does not explicitly teach the usage of a Flash player. Official notice is taken that it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to implement the browser to utilize a Flash player because Flash players were old and well known in the art. One of ordinary skill in the art would have been motivated to use a Flash player because of the common usage within web browsing.

67. Regarding claim 82, Tuli teaches a user device including a rendering entity (col. 1, line 66 – col. 2, ll. 4, operating system with a mini-browser) but does not explicitly teach "browser in a plug-in to said browser is present." However, in related art, Rubin teaches on this limitation wherein a portable device is equipped with plug-in installed within a user device's web browser (col. 7, ll. 18-21). One of ordinary skill in the art at the time of the applicant's invention would have found it obvious to utilize the plug-in as taught by Rubin in combination with the portable device web browser taught by Tuli. One of ordinary skill would have been motivated to combine Rubin with Tuli because plug-ins are auxiliary programs added to web browsers that provide them with new functionality (Rubin, col. 7, ll. 21-23).

68. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Tuli and Davis in view of Russell (2002/0069420).

69. Regarding claim 11, Tuli does not expressly disclose the step of authenticating said request, said steps of compiling and transmitting are only performed if said step of authenticating is successful, different types of authenticating are provided for different types of content or for each item of content. Russell teaches on this aspect wherein a network may authenticate a user's request to download content and that if that authentication fails, the server will not allow the user to download the content (par. 94, lines 1-10). Tuli and Russell are analogous art because they are both from the same field of endeavor of content delivery. At the time of invention it would have been obvious to a person of ordinary skill in the art to allow Tuli's invention to authenticate requests for content and to deny delivery of the content if the request does not pass authentication, as taught by Russell. The motivation for doing so would have been to ensure that the user making the request is authorized to access the content (par. 91, lines 6-7). Therefore it would have been obvious to combine Russell with Tuli and Davis for the benefit of authorized access to obtain the invention as specified in claim 11.

70. Claims 14-17, 19, 20 and 74-75 are rejected under 35 U.S.C. 103(a) as being unpatentable over Tuli in view of Wagner (US 6,085,224).

71. Regarding claim 14, Tuli teaches a method for providing content, comprising the steps of receiving a request for particular content, said request is received at a server (col. 2, ll. 5-13, receive request at web server); in response to said request, accessing first code associated with said particular content (col. 2, ll. 9-11, retrieve HTML), said first code includes a mark-up language description and a scripting language description (col col. 2, ll. 9-11, retrieve HTML).

Tuli does teach compiling HTML to create executable code that implements a user interface that provides access to said particular content (col. 2, lines 10-13), said step of compiling is performed at said server in response to said request (col. 2, lines 10-13); and transmitting said executable code from said server to a client (Tuli, col. 2, ll. 40-42, data is sent back to user device). Tuli teaches the compilation of HTML or JAVA but does not explicitly teach the compilation a combination of both to create combined executable code. The combination of a markup language code and a scripting language description is deemed as common in the art as evidenced by Wagner in column 15, line 61 – column 16, line 15 wherein Wagner teaches the use of embedded commands used in an HTML file to include javascript or visual basic script languages. It would have been obvious to one of ordinary skill in the art to include scripting languages embedded within a markup language file as taught by Wagner. One of ordinary skill in the art would have been motivated to embed script languages due to the common use of scripting in markup language files.

72. Regarding claim 15, Tuli discloses that said request is from said client (col. 2, ll. 9-10).

73. Regarding claim 16, Tuli discloses that:

said particular content includes data (col. 2, lines 10-11 and

said data is compiled to executable code during said step of compiling (col. 2, 10-13, browser translator).

74. Regarding claim 17, Tuli teaches the use of a browser translator to process web data (col. 1, ll. 32-36) but does not explicitly teach the compiling into ActionScript. Official notice is taken that it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to implement the browser translator to handle compilation into ActionScript because

ActionScript was old and well known in the art. One of ordinary skill in the art would have been motivated to use ActionScript because of the common usage within web browsing.

75. Regarding claim 19, Tuli discloses said markup language description includes elements which are identified by markup language tags, at least one of said elements provides a script source of said scripting language description (col. 2, ll. 9-10, HTML).

76. Regarding claim 20, Tuli teaches the steps of:

accessing media content, said particular content includes said media content (col. 2, ll. 9-11, retrieve HTML);

transforming said media content to an accepted format, said transforming is separate from said compiling (Tuli, col. 2, ll. 25-29);

providing a reference to said transformed media content in said executable code (col. 2, lines 47-50); and

transmitting said transformed media content with said executable code from said server to said client for execution by said client, where, during the execution at said client, when said reference is reached, said client renders said transformed media content (Tuli, col. 2, ll. 40-42, data is processed at user device).

77. Regarding claim 74, Tuli and Wagner teach a method wherein at least one of said elements of said markup language description instantiates a class defined in the scripting language description (Wagner in column 15, line 61 – column 16, line 15).

78. Regarding claim 75, Tuli and Wagner teach a method wherein said scripting language description extends a class defined in said markup language description (Wagner in column 15, line 61 – column 16, line 15).

79. Claims 68 and 79 is rejected under 35 U.S.C. 103(a) as being unpatentable over Tuli and Rubin in view of Davis.

80. Regarding claim 68, Tuli teaches the accessing of a mark-up language description but not does not explicitly teach "accessing said data at said external data source based on said one or more source files which define said connection to said external data source, said server performs said accessing." However, in related art, Davis teaches on this aspect wherein a client device can send a request to a server for secondary content (col. 5, lines 54-58) and that the secondary content can be from an external data source (abstract, line 7). Tuli and Davis are analogous art because they are both from the same field of endeavor of computer systems. At the time of invention, it would have been obvious to one of ordinary skill in the art that Davis's method of calling an application from a previously downloaded webpage could be used with Tuli's method of compiling code at a server rather than at the client. After Davis's webpage is downloaded with Tuli's system, Davis's webpage would call the secondary application and Tuli's system would then proceed to locate and compile that secondary application for presentation to the client. The motivation for doing so would have been to allow the users of Tuli's system to be able to utilize content of the type described in Davis on a thin-client device (col. 1, ll. 14-18). Therefore it would have been obvious to combine Davis with Tuli for the benefit of utilizing more complex content on a thin-client device.

81. Regarding claim 79, Tuli teaches the accessing of a mark-up language description but not does not explicitly teach "accessing said data at said external data source based on said one or more source files which define said connection to said external data source, said server performs said accessing." However, in related art, Davis teaches on this aspect wherein a client device can

send a request to a server for secondary content (col. 5, lines 54-58) and that the secondary content can be from an external data source (abstract, line 7). Tuli and Davis are analogous art because they are both from the same field of endeavor of computer systems. At the time of invention, it would have been obvious to one of ordinary skill in the art that Davis's method of calling an application from a previously downloaded webpage could be used with Tuli's method of compiling code at a server rather than at the client. After Davis's webpage is downloaded with Tuli's system, Davis's webpage would call the secondary application and Tuli's system would then proceed to locate and compile that secondary application for presentation to the client. The motivation for doing so would have been to allow the users of Tuli's system to be able to utilize content of the type described in Davis on a thin-client device (col. 1, ll. 14-18). Therefore it would have been obvious to combine Davis with Tuli for the benefit of utilizing more complex content on a thin-client device.

Response to Arguments

82. Applicant's arguments, see Remarks, filed 01 February 2008, with respect to the rejection(s) of claim(s) 33-36, 55-58, 60-62, 64-66, 77 and 78 under 35 USC 102(b) as being anticipated by Wu (US 5,987,256), claims 1, 3-10, 13, 21-24, 26-32, 37-47, 51-54, 67, 69, 70 and 79 under 35 USC 103(a) as being unpatentable over Wu in view of Davis (US 6,643,696), claim 11 under 35 USC 103(a) as being unpatentable over Wu in view of Russell (US 2002/0069420), claims 14-17, 19, 20, 48-50, 74 and 75 under 35 USC 103(a) as being unpatentable over Wu in view of Wagner (US 6,085,224) and claims 73, 76 and 80 under 35 USC 103(a) as being unpatentable over Wu and Davis in view of Ausems (US 2003/0013483) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon

further consideration, a new ground(s) of rejection is made in view of Tuli (US US 7,068,381 B1), Davis et al. (US 6,643,696 B2), Rubin (US 6,701,522), Russell (US 2002/0069420) and Wagner (US 6,085,224) as set forth in the above rejections.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Benjamin Ailes whose telephone number is (571)272-3899. The examiner can normally be reached Monday-Friday, 5:30-8:30AM, 1:00-6:00PM, IFP Hoteling schedule.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Andrew Caldwell can be reached on 571-272-3868. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

BAA

/Andrew Caldwell/
Supervisory Patent Examiner, Art Unit 2142